



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

09/901,368

07/09/2001

Nithyalakshmi Sampathkumar

MS180587.1

6483

27195

7590

05/01/2008

AMIN. TUROCY & CALVIN, LLP  
24TH FLOOR, NATIONAL CITY CENTER  
1900 EAST NINTH STREET  
CLEVELAND, OH 44114

EXAMINER

HILLERY, NATHAN

ART UNIT

PAPER NUMBER

2176

NOTIFICATION DATE

DELIVERY MODE

05/01/2008

ELECTRONIC

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

docket1@thepatentattorneys.com  
hholmes@thepatentattorneys.com  
osteuball@thepatentattorneys.com

<b>Office Action Summary</b>	<b>Application No.</b> 09/901,368	<b>Applicant(s)</b> SAMPATHKUMAR ET AL.	
	<b>Examiner</b> NATHAN HILLERY	<b>Art Unit</b> 2176	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) ☒ Responsive to communication(s) filed on 31 January 2008.
- 2a) ☒ This action is **FINAL**.                      2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) ☒ Claim(s) 1-19 and 38 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-19 and 38 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All    b) ☐ Some \*    c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

- |  |   |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892)                     | 4) <input type="checkbox"/> Interview Summary (PTO-413)           |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____                                      |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)          | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date _____  | 6) <input type="checkbox"/> Other: _____                          |

### DETAILED ACTION

1. This action is responsive to communications: Amendment filed on 1/31/08.
2. Claims 1 – 19 and 38 are pending in the case. Claims 1, 19 and 38 are independent.

### ***Claim Rejections - 35 USC § 103***

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically taught or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

4. Claim 1, 3, 4, 5 and 19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kuznetsov (US 6772413 B2).
5. **Regarding independent claim 1**, Kuznetsov teaches that the generalized system can be applied to the growing problem of integrating disparate or incompatible computer systems, file formats, network protocols, or other machine data. This applies whether the data is recorded in a storage device, preserved in temporary memory, or transmitted over a network. This approach allows many more formats and protocols to be accommodated flexibly while preserving the performance and simplification advantages (Column 9, lines 25 – 33), which meet the limitation of **a memory configured to receive and store one or more input XML items**.

Kuznetsov teaches that to transform an input XML vocabulary to another (output) XML vocabulary, the XSLT translator processor must parse the transform, parse the source data, walk the two parse trees to apply the transform, and finally output the data

into a stream (Column 14, lines 51 – 59), which meets the limitation of **a transformer that transforms one or more input XML items in a first format to one or more transformed XML items in one or more second XML formats.**

Kuznetsov teaches that any number of translators can be implemented simultaneously, such that an entire set (or selected subset) of packets can be translated during runtime (Column 13, line 66 – column 14, line 1), which meets the limitation of **an output manager that facilitates at least one of selectively pulling and pushing a subset of the one or more input XML items.**

Kuznetsov does not explicitly say that **the subset of the one or more XML items is less than the whole one or more input XML items.**

However Kuznetsov teaches an entire set or selected subset (Column 13, line 66 – Column 14, line 1) as two different entities. Thus, it would have been to a person of ordinary skill in the art to try translating a selected subset that is less than the entire set in an attempt to provide an improved translation, as a person with ordinary skill has good reason to pursue the known options within his or her technical grasp. In turn, because the subset as claimed has the properties predicted by the prior art, it would have been obvious to translate a selected subset that is less than the entire set.

6. **Regarding dependent claim 3**, Kuznetsov teaches that a data translator compiler is adapted for using the XSL stylesheet as its input. The data translator compiler then generates executable machine code that operates as a run-time translator between the source XML and the target XML (Column 14, line 60 – Column

15, line 2), which meets the limitation of **a compiler that compiles one or more style sheets and produce one or more actions that can be employed by the transformer in processing associated with transforming the one or more input XML items.**

7. **Regarding dependent claim 4**, Kuznetsov teaches that an implementation according the present invention may also incorporate predefined functions, or references to external functions that can be called at runtime, according to the needs of the translator, as generated by the translator compiler engine (Column 14, lines 23 – 27), which meets the limitation of **the compiler resolves one or more external references in the one or more style sheets.**

8. **Regarding dependent claim 5**, Kuznetsov teaches that whether the data is recorded in a storage device, preserved in temporary memory, or transmitted over a network, the approach allows many more formats and protocols to be accommodated flexibly while preserving the performance and simplification advantages (Column 9, lines 28 – 33), which meets the limitation of **the input XML items are input from one or more data stores.**

9. **Regarding independent claim 19**, Kuznetsov teaches that the generalized system can be applied to the growing problem of integrating disparate or incompatible computer systems, file formats, network protocols, or other machine data. This applies whether the data is recorded in a storage device, preserved in temporary memory, or

Art Unit: 2176

transmitted over a network. This approach allows many more formats and protocols to be accommodated flexibly while preserving the performance and simplification advantages (Column 9, lines 25 – 33), which meet the limitation of **a memory configured to receive and store one or more input XML items.**

Kuznetsov teaches that to transform an input XML vocabulary to another (output) XML vocabulary, the XSLT translator processor must parse the transform, parse the source data, walk the two parse trees to apply the transform, and finally output the data into a stream (Column 14, lines 51 – 59), which meets the limitation of **a transforming component that transforms an input XML item from a first format to a transformed XML item in one or more second XML formats.**

Kuznetsov teaches that any number of translators can be implemented simultaneously, such that an entire set (or selected subset) of packets can be translated during runtime (Column 13, line 66 – column 14, line 1), which meets the limitation of **an output managing component that facilitates at least one of selectively pulling and pushing a subset of the input XML item, the subset of the one or more XML items is less than the whole input XML item.**

Kuznetsov teaches that a data translator compiler is adapted for using the XSL stylesheet as its input. The data translator compiler then generates executable machine code that operates as a run-time translator between the source XML and the target XML (Column 14, line 60 – Column 15, line 2), which meets the limitation of **a compiling component that compiles a style sheet and that produces one or more actions**

**that can be employed by the transforming component in processing associated with transforming the input XML item.**

Kuznetsov teaches that whether the data is recorded in a storage device, preserved in temporary memory, or transmitted over a network, the approach allows many more formats and protocols to be accommodated flexibly while preserving the performance and simplification advantages (Column 9, lines 28 – 33) and that the optimization options comprise first optimization pass, which generates intermediate format, and second optimization pass (Column 16, lines 49 – 54), which meets the limitation of **an input abstracting component that presents input XML items stored in one or more different representations to the transforming component in a common representation.**

Kuznetsov teaches that as currently specified by the Worldwide Web Consortium, there are three major components in an XSL processor: XSLT, the transformation engine; Xpath, the node selection and query module; and Formatting Objects, the formatting and end-user presentation layer specification. XML-to-XML data translation is primarily concerned with the first two modules (Column 14, lines 33 – 39), which meets the limitation of **a node selection abstracting component that dynamically constructs a subset of input XML items from a set of input XML items, the subset of input XML items are responsive to a query.**

10. Claim 2 is rejected under 35 U.S.C. 103(a) as being unpatentable over Kuznetsov (US 6772413 B2) as applied to claim 1 above and further in view of Omoigui (US 20030126136 A1).

11. **Regarding dependent claim 2, Kuznetsov does not explicitly teach that the transformer comprises an action frame stack that holds one or more actions, an event state machine that tracks state associated with transforming the one or more input XML items and an event processor that receives events generated in processing the one or more actions stored in the action frame stack.**

However, Omoigui teaches that the system provides support for authentication, authorization, auditing, data privacy, data integrity, availability, and non-repudiation by employing standards such as WS-Security. WS-Security provides a platform for security with XML Web Service applications using standards in the XML Web Service protocol stack. This includes encrypting method calls from clients, support for digital signatures, authenticating the calling user before granting access to an Agency's Semantic Network and XML Web Service methods, etc. (paragraph block 0367), which meets the limitation **of the transformer comprises an action frame stack that holds one or more actions, an event state machine that tracks state associated with transforming the one or more input XML items and an event processor that receives events generated in processing the one or more actions stored in the action frame stack.**

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of Kuznetsov with the invention of Omoigui because such a combination would provide the readers of Kuznetsov with *an integrated and seamless implementation framework and resulting medium for knowledge retrieval, management, delivery and presentation* (paragraph block 0071).



12. Claims 6 – 18 and 38 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kuznetsov (US 6772413 B2) as applied to claim 1 above and further in view of ADO.NET (English translation).

13. **Regarding dependent claim 6**, Kuznetsov does not explicitly teach that **an input abstractor that exposes data stored in the one or more data stores in a common representation**.

ADO.NET teach that an XPathNavigator is created to abstract data from the xml data set via an XPathNodeIterator by employing a loop (p 19), which meets the limitation of **an input abstractor that exposes data stored in the one or more data stores in a common representation**.

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of Kuznetsov with those of ADO.NET because such a combination would provide the users of Kuznetsov with the benefit of explicit implementation of XPath via source code.

14. **Regarding dependent claim 7**, Kuznetsov does not explicitly teach that **the input abstractor abstracts a reference to a node within an XPath document**.

ADO.NET teach that an XPathNavigator is created to abstract data from the xml data set via an XPathNodeIterator (p 19), which meets the limitation of **the input abstractor abstracts a reference to a node within an XPath document**.

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of Kuznetsov with those of ADO.NET because such

a combination would provide the users of Kuznetsov with the benefit of explicit implementation of XPath via source code.

15. **Regarding dependent claims 8**, Kuznetsov does not explicitly teach that **the input abstractor exposes the data stored in the one or more data stores as a data model and info set**.

ADO.NET teach that an XPathNavigator is created to abstract data from the xml data set (p 19), which meets the limitation of **the input abstractor exposes the data stored in the one or more data stores as a data model and info set**.

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of Kuznetsov with those of ADO.NET because such a combination would provide the users of Kuznetsov with the benefit of explicit implementation of XPath via source code.

16. **Regarding dependent claim 9**, Kuznetsov does not explicitly teach that **the input abstractor provides a cursor model over data stored in a data store to facilitate presenting a stream of nodes to the transformer**.

ADO.NET teach that an XPathNavigator is created to abstract data from the xml data set and sends the data to an XSLT (p 19), which meets the limitation of **the input abstractor provides a cursor model over data stored in a data store to facilitate presenting a stream of nodes to the transformer**.

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of Kuznetsov with those of ADO.NET because such a combination would provide the users of Kuznetsov with the benefit of explicit implementation of XPath via source code.

17. **Regarding dependent claim 10**, Kuznetsov does not explicitly teach that **the input abstractor provides a virtual node that can be employed to traverse the stream of nodes**.

ADO.NET teach that an XPathNavigator is created to abstract data from the xml data set (p 19), which meets the limitation of **the input abstractor provides a virtual node that can be employed to traverse the stream of nodes**.

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of Kuznetsov with those of ADO.NET because such a combination would provide the users of Kuznetsov with the benefit of explicit implementation of XPath via source code.

18. **Regarding dependent claim 11**, Kuznetsov does not explicitly teach that **the input abstractor is an XPathNavigator**.

ADO.NET teach that an XPathNavigator is created to abstract data from the xml data set (p 19), which meets the limitation of **the input abstractor is an XPathNavigator**.

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of Kuznetsov with those of ADO.NET because such a combination would provide the users of Kuznetsov with the benefit of explicit implementation of XPath via source code.

19. **Regarding dependent claim 12**, Kuznetsov does not explicitly teach that **a node selection abstractor that dynamically constructs a subset of input XML items from a set of input XML items, the subset of input XML items are responsive to a query.**

ADO.NET teach that SQL is used to query xml items and store them to an XML data set and that each node in the xml dataset is visited by employing an XpathNodeIterator (pp 18 – 19), which meets the limitation of **a node selection abstractor that dynamically constructs a subset of input XML items from a set of input XML items, the subset of input XML items are responsive to a query.**

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of Kuznetsov with those of ADO.NET because such a combination would provide the users of Kuznetsov with the benefit of explicit implementation of XPath via source code.

20. **Regarding dependent claim 13**, Kuznetsov does not explicitly teach that **the node selection abstractor facilitates navigating the subset of input XML items.**

ADO.NET teach that each node in the xml dataset is visited by employing an XpathNodeIterator (pp 18 – 19), which meets the limitation of **the node selection abstractor facilitates navigating the subset of input XML items**.

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of Kuznetsov with those of ADO.NET because such a combination would provide the users of Kuznetsov with the benefit of explicit implementation of XPath via source code.

21. **Regarding dependent claim 14**, Kuznetsov does not explicitly teach that **the node selection abstractor is an XpathNodeIterator**.

ADO.NET teach that each node in the xml dataset is visited by employing an XpathNodeIterator (pp 18 – 19), which meets the limitation of **the node selection abstractor is an XpathNodeIterator**.

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of Kuznetsov with those of ADO.NET because such a combination would provide the users of Kuznetsov with the benefit of explicit implementation of XPath via source code.

22. **Regarding dependent claim 15**, Kuznetsov does not explicitly teach that **an optimized data store that stores one or more XML items in a manner that facilitates minimizing processing associated with constructing the subset of input XML items via a query**.

ADO.NET teach that SQL is used to query xml items and store them to an XML data set (pp 18 – 19), which meets the limitation of **an optimized data store that stores one or more XML items in a manner that facilitates minimizing processing associated with constructing the subset of input XML items via a query.**

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of Kuznetsov with those of ADO.NET because such a combination would provide the users of Kuznetsov with the benefit of explicit implementation of XPath via source code.

23. **Regarding dependent claim 16**, Kuznetsov does not explicitly teach that **the optimized data store stores data in a data representation format that facilitates optimizing an Xpath query.**

ADO.NET teach that Xpath document is created and used to store and manipulate the xml data set (pp 18 – 19), which meets the limitation of **the optimized data store stores data in a data representation format that facilitates optimizing an Xpath query.**

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of Kuznetsov with those of ADO.NET because such a combination would provide the users of Kuznetsov with the benefit of explicit implementation of XPath via source code.

24. **Regarding dependent claim 17**, Kuznetsov does not explicitly teach that **the data representation format comprises expanded XML entities, deleted XML declarations and DOM model data converted to Xpath model data.**

ADO.NET teach that Xpath document is created and used to expand the items in the xml data store so that they can be transformed using an XSLT (pp 18 – 19), which meets the limitation of **the data representation format comprises expanded XML entities, deleted XML declarations and DOM model data converted to Xpath model data.**

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of Kuznetsov with those of ADO.NET because such a combination would provide the users of Kuznetsov with the benefit of explicit implementation of XPath via source code.

25. **Regarding dependent claim 18**, Kuznetsov does not explicitly teach that **the optimized data store is an XPathDocument.**

ADO.NET teach that Xpath document is created and used to store and manipulate the xml data set (pp 18 – 19), which meets the limitation of **the optimized data store is an XPathDocument.**

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of Kuznetsov with those of ADO.NET because such a combination would provide the users of Kuznetsov with the benefit of explicit implementation of XPath via source code.

26. **Regarding claim 38**, the claim incorporates substantially similar subject matter as claim 12 and is rejected along the same rationale.

27.

***Response to Arguments***

28. Applicant's arguments filed 1/31/08 have been fully considered but they are not persuasive.

Applicant argues that Kuznetsov does not teach **a transformer that transforms one or more input XML items in a first format to one or more transformed XML items in one or more second XML formats**, because Kuznetsov teaches a plurality of translators and turning one of them off would make claim 1 inoperative (pp 10 – 12).

The office disagrees.

First, the office does not fully understand applicant's arguments. Simply, Kuznetsov teaches that to transform an input XML vocabulary to another (output) XML vocabulary, the XSLT translator processor must parse the transform, parse the source data, walk the two parse trees to apply the transform, and finally output the data into a stream (Column 14, lines 51 – 59).

The Office maintains that Kuznetsov clearly and explicitly teaches transforming XML items. Even if, for the sake of argument, everything applicant says is true, Kuznetsov still meets the claimed language because the claim only requires that the 'transformer' transform one XML item. Therefore, as long as one of the alleged 'translators' of Kuznetsov is responsible for transforming one XML item out of the lot then Kuznetsov meets the claim language.



29. In response to applicant's argument that the examiner's conclusion of obviousness is based upon improper hindsight reasoning (p 12, bottom – p 13, top), it must be recognized that any judgment on obviousness is in a sense necessarily a reconstruction based upon hindsight reasoning. But so long as it takes into account only knowledge which was within the level of ordinary skill at the time the claimed invention was made, and does not include knowledge gleaned only from the applicant's disclosure, such a reconstruction is proper. See *In re McLaughlin*, 443 F.2d 1392, 170 USPQ 209 (CCPA 1971).

30.

31. Applicant argues that Omoigui fails to teach that **the transformer comprises an action frame stack that holds one or more actions, an event state machine that tracks state associated with transforming the one or more input XML items and an event processor that receives events generated in processing the one or more actions stored in the action frame stack** because the action stack can be considered analogous but not equivalent to a queue that can store actions (pp 13 and 14).

The Office disagrees.

First, by applicant's own admission, an action stack can be considered analogous, although not equivalent, to a queue that can store actions while other actions are being processed (p 13, second paragraph). It is respectfully submitted that the 'queue' disclosed by Omoigui is analogous to the claimed action frame stack and thus meets the limitation within the broadest, reasonable interpretation in light of the specification as required under 35 USC 103.

Applicant argues that the ADO.NET reference should not be afforded its rightful date because the date the reference was published or placed on the Internet is unknown (p 15).

The Office disagrees.

As explained to applicant previously, the reference consists of slides that were presented at a conference that was held March 6 – 8, 2001, which antedates applicant's effective filing date. MPEP 2128.01 provides support. Specifically, publicly displayed documents can constitute a "printed publication" even if the duration of display is for only a few days and the documents are not disseminated by copies or indexed in a library or database.

32. Applicant generally argues claims 6 – 10 and 12 – 17 as not being taught because the reference lacks certain key words (pp 11 – 13).

33. Conveniently, applicant fails to mention that claims 11 and 18 provide evidence that claims 6 – 10 and 12 – 17 are met by the reference because ADO.NET teaches a XpathNavigator and XpathDocument, respectively, which is all the claims require.

34.

### ***Conclusion***

35. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to NATHAN HILLERY whose telephone number is (571)272-4091. The examiner can normally be reached on M - F, 10:30 a.m. - 7:00 p.m.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Doug Hutton can be reached on (571) 272-4137. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Art Unit: 2176

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

NH

*/Doug Hutton/*  
Doug Hutton  
Supervisory Primary Examiner  
Technology Center 2100

**Search Notes****Application/Control No.**

09/901,368

**Examiner**

NATHAN HILLERY

**Applicant(s)/Patent under Reexamination**

SAMPATHKUMAR ET AL.

**Art Unit**

2176

**SEARCHED**

Class	Subclass	Date	Examiner
715	513 523 524 514 515	2/18/2005	NH
ABOVE	UPDATED	7/1/2005	NH
ABOVE	UPDATED	1/12/2006	NH
ABOVE	UPDATED	7/5/2006	NH
ABOVE	UPDATED	10/25/2006	NH
ABOVE	UPDATED	4/13/2007	NH
ABOVE	UPDATED	10/25/2007	NH
ABOVE	UPDATED	4/23/2008	NH

**INTERFERENCE SEARCHED**

Class	Subclass	Date	Examiner

**SEARCH NOTES  
(INCLUDING SEARCH STRATEGY)**

	DATE	EXMR
EAST (USPT, USPGPUB, JPO, EPO, Derwent, IBMTD) - See Search History Printout	2/18/2005	NH
EAST (USPT, USPGPUB, JPO, EPO, Derwent, IBMTD) - See Search History Printout	7/1/2005	NH
EAST (USPT, USPGPUB, JPO, EPO, Derwent, IBMTD) - See Search History Printout	1/12/2006	NH
EAST (USPT, USPGPUB, JPO, EPO, Derwent, IBMTD) - See Search History Printout	7/5/2006	NH
EAST (USPT, USPGPUB, JPO, EPO, Derwent, IBMTD) - See Search History Printout	10/25/2006	NH
EAST (USPT, USPGPUB, JPO, EPO, Derwent, IBMTD) - See Search History Printout	4/13/2007	NH
EAST (USPT, USPGPUB, JPO, EPO, Derwent, IBMTD) - See Search History Printout	10/25/2007	NH
EAST (USPT, USPGPUB, JPO, EPO, Derwent, IBMTD) - See Search History Prin	4/23/2008	NH